

UNIVERSIDADE DE SÃO PAULO  
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**Segmentação de tumores em imagens  
de ressonância magnética 3D**

Daniela Gonzalez Favero

MONOGRAFIA FINAL

MAC 499 — TRABALHO DE  
FORMATURA SUPERVISIONADO

Orientador: Prof. Dr. Roberto Marcondes Cesar Junior

Coorientador: Dr. Hugo Neves de Oliveira

Durante o desenvolvimento deste trabalho a autora recebeu auxílio financeiro do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq). O desenvolvimento do projeto conta com apoio FAPESP #2015/22308-2 e #2017/50236-1.

São Paulo  
28 de Janeiro de 2022



# Agradecimentos

*A dúvida deve seguir a convicção como uma sombra.*

— Albert Camus

Agradeço ao professor Roberto e ao Hugo pela orientação neste trabalho, ambos se mostrando sempre dispostos a ensinar e ajudar com o que fosse necessário. Também gostaria de agradecer aos demais professores que tive no decorrer do curso, que me prepararam para este trabalho de conclusão.

Aos meus pais, agradeço pelo apoio emocional e financeiro, pelos cuidados e conselhos para alcançar meus sonhos. Obrigada por sempre estarem ao meu lado. Também agradeço aos meus colegas de curso que acabaram por se tornar amigos para toda a vida, pelas noites de estudo em conjunto e apoio durante o desenvolvimento deste trabalho.

Finalmente, agradeço ao William, meu grande amigo e apoio, e à Maria do Amparo, querida psicóloga, pela ajuda nesses dois últimos anos difíceis de pandemia. A companhia e incentivo de vocês foi essencial.



# Resumo

Daniela Gonzalez Favero. **Segmentação de tumores em imagens de ressonância magnética 3D**. Monografia (Bacharelado). Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2022.

Os avanços em imagens médicas requerem o desenvolvimento de métodos que melhorem a acurácia nos resultados da análise de imagem. Ferramentas automatizadas para detecção de anomalias e anotação das amostras obtidas têm se tornado altamente demandadas. Este trabalho tem como objetivo aplicar visão computacional ao domínio de imagens médicas. A partir de imagens de ressonância magnética do cérebro adquiridas clinicamente, é possível treinar os algoritmos com dados e produzir classes de segmentação para a detecção de tumores. Assim, é possível compreender melhor a mudança e evolução de estruturas cerebrais e possíveis patologias, de modo a gerar diagnósticos e análises tumorais. O desafio se encontra no formato dos dados, que por muitas vezes são ruidosos, ambíguos, escassos e temporalmente esparsos. Além disso, se pretende obter um detector aplicável a mais de uma região do cérebro. Finalmente, o projeto deve cobrir o estudo de ressonâncias magnéticas pediátricas.

**Palavras-chave:** Aprendizado Profundo. Segmentação Semântica. Imagens de Ressonância Magnética Cerebrais.



# Abstract

Daniela Gonzalez Favero. **Tumor segmentation of MRI scans**. Capstone Project Report (Bachelor). Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2022.

The advances in medical imaging require the development of methods that improve the accuracy of the results of image analysis. Automated tools for anomaly detection and sample annotation have been highly demanded. This project's goal is to apply computer vision to the medical imaging domain. From clinically acquired magnetic resonance brain images, it is possible to train the algorithms with data and produce segmentation classes for tumor detection. The challenge lies in the data format, which is often noisy, ambiguous, scarce, and temporally sparse. Furthermore, it is intended to obtain a detector that is applicable to more than a single brain region. Finally, the project should cover the study of pediatric magnetic resonances.

**Keywords:** Deep Learning. Semantic Segmentation. Brain MRIs.





# Lista de Abreviaturas

BraTS	Segmentação de Tumor Cerebral ( <i>Brain Tumor Segmentation</i> )
CE	Entropia Cruzada ( <i>Cross Entropy</i> )
CBICA	Centro para Computação e Análise de Imagens Biomédicas e Análises ( <i>Center for Biomedical Image Computing and Analytics</i> )
CNN	Redes Neurais Convolucionais ( <i>Convolutional Neural Networks</i> )
ED	Edema Peritumoral ( <i>Peritumoral Edema</i> )
ET	Tumor em Crescimento ( <i>Enhancing Tumor</i> )
FCN	Redes Totalmente Convolutivas ( <i>Fully Convolutional Networks</i> )
FeTA	Anotação do Tecido Cerebral de Fetos ( <i>Fetal Brain Tissue Annotation</i> )
GM	Substância Cinzenta ( <i>Grey Matter</i> )
HGG	Gliomas de Alto Grau ( <i>High-Grade Gliomas</i> )
HR3N	<i>HighRes3DNet</i>
LCR	Líquido Cefalorraquidiano
LGG	Gliomas de Baixo Grau ( <i>Low-Grade Gliomas</i> )
MICCAI	Computação de imagens médicas e intervenções assistidas por computador ( <i>Medical Image Computing and Computer Assisted Interventions</i> )
MLP	Perceptron Multicamadas ( <i>MultiLayer Perceptron</i> )
MRI	Imagem de Ressonância Magnética ( <i>Magnetic Resonance imaging</i> )
NCR	Necrótico ( <i>Necrotic</i> )
NET	Tumor Não em Crescimento ( <i>Non-Enhancing Tumor</i> )
NIFTI	Iniciativa de Tecnologia de Informática de Neuroimagem ( <i>Neuroimaging Informatics Technology Initiative</i> )
ReLU	Unidade Linear Retificada ( <i>Rectified Linear Unit</i> )
RoI	Região de Interesse ( <i>Region of Interest</i> )
SDN	<i>SkipDenseNet</i>
WM	Substância Branca ( <i>White Matter</i> )

## Lista de Símbolos

- $\gamma$  Decaimento da taxa de aprendizado a cada determinada quantia de épocas
- $\sigma$  Desvio padrão

# Lista de Figuras

3.1	A arquitetura da U-Net 3D. As caixas azuis representam as <i>feature maps</i> e o número de canais está descrito sobre cada caixa. Figura extraída de (ÇIÇEK <i>et al.</i> , 2016). . . . .	10
3.2	A arquitetura da V-Net. Figura extraída de (MILLETARI <i>et al.</i> , 2016). . . . .	11
3.3	A arquitetura da HighRes3DNet. Figura extraída de (LI <i>et al.</i> , 2017). . . . .	12
3.4	Exemplo de aplicação da SkipDenseNet sobre MRIs cerebrais de contraste T1 e T2. Figura extraída de (BUI <i>et al.</i> , 2017). . . . .	13
3.5	Arquitetura da ResNet18, utilizada para codificação na Med3D. Note que a camada totalmente conectada (FC) no esquema é substituída pelas camadas de decodificação. Figura extraída de (RAMZAN <i>et al.</i> , 2020). . . . .	13
3.6	<i>Pipeline</i> proposta para a segmentação do conjunto de dados BraTS. A MRI inteira $x_G$ é submetida à rede generalista $\mathcal{G}$ , gerando a predição $\hat{y}_G$ . Então, a RoI $x_S$ obtida a partir da saída de $\mathcal{G}$ é submetida à rede especialista $\mathcal{S}$ , gerando a predição $\hat{y}_S$ . . . . .	15
3.7	<i>Pipeline</i> proposta para a segmentação do conjunto de dados FeTA. . . . .	16
4.1	Sub-regiões do glioma. Em (A) é possível ver WT (amarelo) em T2-FLAIR, em (B) o TC (vermelho) visível em T2 e em (C) o ET (em azul) e o tecido necrótico, ambos visíveis em T1Gd. À direita, cada região segmentada é combinada na MRI. Figura extraída de (BAKAS, REYES <i>et al.</i> , 2018). . . . .	18
4.2	Regiões cerebrais do feto a serem segmentadas. LCR externo em verde escuro, GM em amarelo, WM em marrom, ventrículos em azul, cerebelo em vermelho, <i>deep</i> GM em laranja e tronco cerebral em verde claro. Figura extraída de (PAYETTE <i>et al.</i> , 2021). . . . .	19
5.1	Matriz de confusão da primeira <i>pipeline</i> sobre os dados do BraTS, utilizando a arquitetura Med3D. . . . .	24
5.2	Matrizes de confusão da segunda <i>pipeline</i> sobre os dados do BraTS. . . . .	25

5.3	Segmentação das sub-regiões do glioma. As colunas indicam, em ordem, fatias sagitais da imagem $X$ , a região de interesse $X_{ROI}$ , o rótulo da região $Y_{ROI}$ e as predições $\hat{Y}$ (da primeira <i>pipeline</i> ), $\hat{Y}_G$ (da rede generalista) e $\hat{Y}_S$ (da rede especialista) . . . . .	25
5.4	Matriz de confusão da primeira <i>pipeline</i> sobre os dados do FeTA, utilizando a arquitetura U-Net. . . . .	27
5.5	Matrizes de confusão da segunda <i>pipeline</i> sobre os dados do FeTA. . . . .	28
5.6	Segmentação das sub-regiões do cérebro fetal. As colunas indicam, em ordem, fatias sagitais da imagem $X$ , o rótulo $Y$ e as predições $\hat{Y}$ ; uma para cada <i>pipeline</i> . . . . .	29

## Listas de Tabelas

4.1	Configurações que especificam as opções do experimento a ser realizado. . . . .	19
4.2	Hiperparâmetros do treinamento. . . . .	19
4.3	Hiperparâmetros dos experimentos que dizem respeito ao otimizador. . . . .	20
4.4	Configurações dos experimentos com respeito ao <i>dataset</i> . . . . .	20
5.1	Métricas <i>DSC</i> e <i>IoU</i> da primeira <i>pipeline</i> executada sobre os dados do BraTS. . . . .	23
5.2	Métricas <i>DSC</i> e <i>IoU</i> da segunda <i>pipeline</i> executada sobre os dados do BraTS, comparando as execuções de $G$ e $S$ . . . . .	24
5.3	Métricas <i>DSC</i> e <i>IoU</i> da primeira <i>pipeline</i> executada sobre os dados do FeTA. . . . .	26
5.4	Métricas <i>DSC</i> e <i>IoU</i> da segunda <i>pipeline</i> executada sobre os dados do FeTA, comparando as execuções de $G$ e $S$ . . . . .	27

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Motivação . . . . .	1
1.2	Objetivos . . . . .	3
1.3	Organização do trabalho . . . . .	3
<b>2</b>	<b>Conceitos sobre Segmentação de Imagens</b>	<b>5</b>
2.1	Redes Neurais Convolucionais . . . . .	6
2.2	O Treinamento da Rede . . . . .	7
2.3	Aprendizado profundo para segmentação volumétrica . . . . .	7
<b>3</b>	<b>Metodologia</b>	<b>9</b>
3.1	Arquiteturas de Redes Neurais . . . . .	9
3.1.1	U-Net 3D . . . . .	9
3.1.2	V-Net . . . . .	10
3.1.3	HighResNet 3D . . . . .	11
3.1.4	SkipDenseNet 3D . . . . .	12
3.1.5	Med3D . . . . .	12
3.2	Funções de Perda . . . . .	13
3.2.1	Entropia Cruzada . . . . .	14
3.2.2	<i>Dice Loss</i> . . . . .	14
3.3	Redes Especialistas . . . . .	14
3.4	Testes . . . . .	16
3.5	Implementação . . . . .	16
<b>4</b>	<b>Setup Experimental</b>	<b>17</b>
4.1	Conjuntos de dados . . . . .	17
4.1.1	O conjunto de dados BraTS . . . . .	17
4.1.2	O conjunto de dados FeTA . . . . .	18
4.2	Hiperparâmetros . . . . .	19

4.3	Métricas . . . . .	21
4.3.1	Coeficiente <i>Dice</i> . . . . .	21
4.3.2	Índice de Jaccard . . . . .	21
4.3.3	Matriz de confusão . . . . .	21
<b>5</b>	<b>Resultados</b>	<b>23</b>
5.1	Execução sobre os dados do BraTS . . . . .	23
5.2	Execução sobre os dados do FeTA . . . . .	26
<b>6</b>	<b>Conclusão</b>	<b>31</b>
	<b>Referências</b>	<b>33</b>

# Capítulo 1

## Introdução

### 1.1 Motivação

O estudo de imagens de ressonância magnética (ou MRI, do inglês “*Magnetic Resonance Imaging*”) é essencial para compreender o desenvolvimento cerebral de fetos e crianças, além da possível manifestação de patologias em diversas regiões do cérebro em pacientes de todas as idades. No entanto, a anotação dessas imagens feita por radiologistas se mostra custosa e frequentemente ambígua, de modo que a possibilidade de automatização desta tarefa é interessante para tais estudos clínicos. As demandas médicas incluem localização e segmentação das regiões de interesse, além da rotulação de patologias nessas estruturas (DESPOTOVIĆ *et al.*, 2015).

Uma das regiões de interesse são as células gliais, que são responsáveis pelo suporte dos neurônios e funcionamento saudável do sistema nervoso. Os gliomas, tumores que atacam essas células, são uma das patologias cerebrais mais comuns. Esses tumores podem ter diferentes graus de agressividade, prognóstico variável e várias sub-regiões histológicas heterogêneas. Essa heterogeneidade intrínseca dos gliomas também impacta nos atributos das imagens obtidas (como aparência e formato), refletindo as propriedades biológicas variáveis dos tumores. Por conta dessa aparência e forma altamente heterogêneas, a segmentação de tumores cerebrais em MRIs multimodais é uma das tarefas mais desafiadoras em análise de imagens médicas.

Há um crescimento na literatura de algoritmos computacionais abordando essa importante tarefa. Atualmente, *datasets* para desenvolver e testar tais algoritmos são limitados. Por isso, o desafio BraTS (*Multimodal Brain Tumor Segmentation*) de 2019<sup>1</sup> (MENZE *et al.*, 2014; BAKAS, REYES *et al.*, 2018; BAKAS, AKBARI *et al.*, 2017), que faz parte da conferência anual da MICCAI Society<sup>2</sup> (organização profissional de *Medical Image Computing and Computer Assisted Interventions*), disponibilizou um *dataset* amplo com definições das sub-regiões relevantes de tumores. As peculiaridades deste conjunto de dados serão detalhadas na Subseção 4.1.1.

---

<sup>1</sup> <https://www.med.upenn.edu/cbica/brats2019/tasks.html>

<sup>2</sup> <http://www.miccai.org>

Outra tarefa interessante é a de segmentar MRIs de fetos. Isso porque doenças congênitas são uma das principais causas de mortalidade neonatal. Assim, a utilização de ressonância magnética para imagens de fetos têm se tornado uma ferramenta útil para quando as imagens de ultrassom não são claras o suficiente para o diagnóstico. Em especial, MRIs cerebrais de feto têm se mostrado importantes para a detecção de doenças como lissencefalia e anomalias no corpo caloso (CARDENAS *et al.*, 2020).

A morfologia cerebral dos fetos pode se modificar de forma complexa e rápida, por isso cresce a importância de se obter segmentação de estruturas específicas mais rapidamente. Clinicamente, é relevante estudar formato e volume de estruturas como o córtex, cerebelo, tronco cerebral, substância branca e os espaços que contêm líquido cefalorraquidiano. MRIs dessas regiões podem revelar diversas patologias e doenças congênitas.

Parte do desafio em segmentar as MRIs fetais consiste no fato das estruturas serem distintas do cérebro de um adulto, além das difíceis condições de obtenção das imagens: os fetos não são sedados no momento da captura, então as MRIs costumam ter artefatos de movimento ou borrões. Apesar dessas imagens serem submetidas a processamentos que remove parte dos artefatos, nem sempre é possível removê-los por completo. Também há a questão de escassez: existem poucos conjuntos de dados de MRIs fetais, ainda mais abertas. A partir disto, o desafio FeTA<sup>3</sup> de 2021, também parte da MICCAI, disponibilizou um *dataset* com MRIs de fetos, anotados como saudáveis ou patológicos. Esse conjunto de dados será melhor descrito na Subseção 4.1.2.

A partir dos *datasets*, se inicia o desafio do processamento dessas MRIs. A segmentação de imagens utilizando aprendizado profundo é majoritariamente feita a partir da arquitetura de Redes Neurais Convolucionais (CNNs) (KRIZHEVSKY *et al.*, 2012), que são redes neurais compostas por camadas de entrada, camadas *hidden* e camadas de saída, das quais algumas camadas são convolucionais. Na segmentação de tumores, CNNs são utilizadas para fazer rotulação densa, ou seja, para determinar uma classe por voxel da imagem. Para que o aprendizado computacional seja efetivo, é necessária uma quantidade considerável de dados de entrada para treinamento e teste. No entanto, no contexto médico, a obtenção das MRIs não é simples e nem realizada em grandes quantidades. Pelo fato dos dados precisarem de rotulações por profissionais altamente especializados, a aquisição das imagens e avaliação do resultados, feita por um grupo limitado de neuro-radiologistas, podem adicionar complexidade à tarefa.

Outro desafio computacional a ser enfrentado, é o fato de que o projeto se encaixa no cenário *small-data*, portanto apresenta conjuntos de dados limitados, porém legíveis por seres humanos (ao contrário de *big-data*). Essa escassez de dados exige especificidade dos modelos construídos, e uma melhor compreensão do *dataset* em questão. Já sob o ponto de vista da dimensionalidade dos dados, é importante ressaltar que as MRIs são imagens 3D, o que aumenta a complexidade do processamento computacional, e faz surgir a necessidade de modelos que lidem com a tridimensionalidade e obtenham o aprendizado mais preciso possível a partir delas.

Os desafios computacionais apontados e a demanda da neuro-oncologia são os maiores motivadores deste projeto. Será particularmente interessante atender a essas importantes

---

<sup>3</sup> <https://feta-2021.grand-challenge.org/>



necessidades clínicas, e facilitar o acompanhamento de tumores em pacientes pediátricos sem que sejam necessárias anotações manuais para a análise das MRIs obtidas.

## 1.2 Objetivos

O objetivo geral deste projeto é o desenvolvimento de algoritmos de segmentação de tumores e de regiões que apresentem condições patológicas em ressonâncias magnéticas.

Dentro do escopo do objetivo geral, este projeto visa alcançar os seguintes objetivos específicos

1. Analisar os conjuntos de dados a serem utilizados, compreendendo suas peculiaridades e seus impactos na construção dos modelos de segmentação;
2. Desenvolver e aplicar arquiteturas de redes neurais sobre as MRIs;
3. Medir a acurácia dos modelos desenvolvidos, comparando os resultados;
4. Investigar a generalização das arquiteturas: entender o quão útil é estender modelos aplicáveis ao *dataset* do BraTS para as imagens do FeTA.

## 1.3 Organização do trabalho

No capítulo 2 desta monografia, será desenvolvida uma contextualização dos conceitos de segmentação automática de imagens utilizando redes neurais. No próximo capítulo, o 3º, será descrito a *pipeline* da metodologia desenvolvida para o treinamento e validação dos modelos. No capítulo 4, há uma apresentação dos resultados obtidos e suas análises. Finalmente, no último capítulo, o 5º, serão expostas as conclusões finais sobre os resultados obtidos e o trabalho em geral.



## Capítulo 2

# Conceitos sobre Segmentação de Imagens

O campo de análise de imagens cresceu rapidamente nas últimas décadas. Assim, o problema de segmentar regiões em imagens e atribuir significado a essas regiões aparece com alta frequência. No contexto de imagens médicas, a segmentação de órgãos e regiões de interesse vem ganhado importância. Inicialmente, eram utilizados métodos clássicos de segmentação (oriundos da teoria de processamento de imagens) adaptados com técnicas de aprendizado de máquina não-supervisionado. Esses métodos, conhecidos como algoritmos de *shallow learning*, utilizam algoritmos baseados em regiões de contexto (POHLE e TOENNIES, 2001) ou clusterização (DHANACHANDRA *et al.*, 2015), por exemplo. Posteriormente, outros algoritmos de *shallow learning* passaram a ser amplamente utilizados, principalmente com a popularização do *machine learning*. Algoritmos de aprendizado de máquina supervisionado como *feature extraction* passam a ser aplicados nesse tipo de tarefa (BOUGACHA *et al.*, 2018; HATAMI *et al.*, 2019; W. CHEN *et al.*, 2017).

De um modo geral, algoritmos de *machine learning* funcionam buscando se aproximar de uma função alvo que mapeia os dados de entrada aos dados de saída. A estimativa é dada a partir de um vetor de pesos que, combinado aos dados de entrada, gera um vetor de saída próximo das saídas esperadas (obtidas pela função alvo). A medida de proximidade entre os dados de saída originais e a saída obtida pelo algoritmo é definida por função de perda que, especialmente em algoritmos de regressão linear, é utilizada para avançar iterativamente no espaço de busca e otimizar a solução.

Esses métodos clássicos acabam gerando diversos problemas tais como processamentos locais, geralmente envolvendo *features* muito rasos, além de generalidade e falta de informação semântica. A partir disto, surgem abordagens de aprendizado profundo para a rotulação densa (ou seja, atribuição de um rótulo por pixel na imagem). A tarefa de segmentação então passa a ser submetida a uma rede neural, que consiste em várias camadas de nós interligados. Esses nós serão determinados a partir do problema a ser resolvido, e serão especificados a seguir.

Os algoritmos de redes neurais nascem com os *MultiLayer Perceptrons* (MLPs), utilizando o algoritmo Perceptron (o pioneiro em aprendizado de máquina) como nó da rede

(LECUN *et al.*, 1998). Um nó Perceptron busca atualizar seu vetor de pesos de modo a se aproximar da solução, vetor de pesos este definido pelos nós de saída da camada anterior e nós da camada atual. Assim, o número de pesos a se considerar em uma camada  $i$  é dado pelo número de neurônios da camada  $i - 1$  vezes o número de neurônios da camada  $i$ . Ao aplicar uma rede neural a uma imagem de dimensões  $N \times M$ , é necessário analisar cada pixel (ou seja, a primeira camada recebe  $NM$  entradas) e rotular cada pixel (da mesma forma, a última camada tem  $NM$  saídas). Isso requer  $N^2 M^2$  pesos na rede, o que custaria muito espaço e tempo para o algoritmo obter resultados satisfatórios em imagens de tamanho médio.

## 2.1 Redes Neurais Convolucionais

É a partir dessa ineficiência das MLPs que surge a necessidade de um neurônio com menor número de parâmetros em redes neurais para imagens. A partir do fato de que convoluções podem ser usadas para armazenar informação visual, constrói-se a ideia de utilizar convoluções nos nós da rede. Convoluções são operações que medem a semelhança entre dois sinais, ou seja, para uma imagem elas podem armazenar informação espacial.

Dada uma imagem  $f(\cdot)$  de dimensões  $X \times Y$  e um *kernel*  $h(\cdot)$  de dimensões  $N \times M$ , a convolução 2D é dada por

$$f(x, y) * h(x, y) = \frac{1}{NM} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} f(n, m)h(x - n, y - m), \quad (2.1)$$

para  $x = 0, 1, \dots, X - 1$  e  $y = 0, 1, \dots, Y - 1$ .

As respostas geradas por convoluções serão mais fortes nas áreas das imagens que tiverem maior similaridade com o conteúdo do *kernel*, tornando-se um bom parâmetro para a extração de informações das imagens analisadas. Além disso, o uso de convoluções em redes neurais para imagens permite que o número de parâmetros seja definido apenas pelos tamanhos dos *kernels*, e não pelas dimensões da imagem. Então, surgem as Redes Neurais Convolucionais (CNNs) (KRIZHEVSKY *et al.*, 2012), que funcionam otimizando os pesos do *kernel* convolucional.

Das possíveis camadas utilizadas em CNNs, as mais comuns são as camadas convolucionais, as camadas de *pooling* e as camadas totalmente conectadas (MLPs). As camadas de *pooling* (LECUN *et al.*, 1998) aplicam uma determinada operação sobre uma imagem para diminuir sua resolução, recebendo como parâmetro um *field of view*  $F$  e um *stride*  $S$ .  $F$  define o tamanho do intervalo sobre o qual a operação será aplicada, e  $S$  é o passo da aplicação. As operações de *pooling* costumam ser de média ou máximo, sendo essa conhecida como *max-pooling*.

Camadas de convolução transposta (DUMOULIN e VISIN, 2016) também costumam ser utilizadas para modificar a resolução das imagens. Ao contrário do *pooling*, a convolução transposta aumenta a resolução dos volumes (realiza *up-sampling*). Essa operação é feita por meio da aplicação da transformação linear definida pela matriz transposta da convolução sobre a imagem.

Outra peça importante para as camadas de uma CNN são as funções de ativação. Essas funções mapeiam o resultado das operações realizadas na camada (como as convoluções) para a saída. A função de ativação mais utilizada em aprendizado profundo para imagens é a ReLU (*Rectified Linear Unit*) (GLOROT *et al.*, 2011), que é definida como a parte positiva de seu argumento:

$$f(x) = \max(0, x), \quad (2.2)$$

sendo  $x$  um neurônio, dado estritamente por um número real. Essa função de ativação evita que fenômenos como o problema *vanishing gradient* tornando a propagação de gradiente mais eficiente. O *vanishing gradient* ocorre em redes que utilizam *backpropagation* (como as arquiteturas *Encoder-Decoder*, que serão melhor explicadas na Seção 3.1), quando o gradiente calculado por cada camada diminui drasticamente na propagação *backward*, de modo a tornar o erro muito pequeno e atrapalhar o desempenho de aprendizado nas camadas finais.

A ReLU se mostra vantajosa em relação a outras funções de ativação (como a sigmoide e a tangente hiperbólica), a medida em que evita problemas como a saturação, que leva os valores de entrada para os extremos do intervalo delimitado pela função, e a falta de sensibilidade: outras funções costumam ser sensíveis apenas a mudanças ao redor do ponto médio do seu intervalo de entrada. Ambos os fenômenos diminuem capacidade de ajuste dos pesos conforme a rede é executada.

## 2.2 O Treinamento da Rede

O treinamento das redes neurais é feito em uma sequência de épocas, cada uma consistindo de uma execução completa da rede sobre o conjunto de treinamento. Uma maneira de atualizar o vetor de pesos comumente utilizada em CNNs é em *mini-batches*. Com *mini-batches*, os pesos do vetor são atualizados diversas vezes em cada época. A quantidade de dados que é utilizada para cada atualização do vetor de pesos é chamada de *batch size* (tamanho do *batch*).

No entanto, é importante notar que a distribuição da entrada de cada camada da rede muda durante o treinamento, afinal os parâmetros das camadas anteriores mudam. Esse fenômeno exige ajustes em hiperparâmetros (como uma taxa de aprendizado menor) que pode causar lentidão na execução da rede. Além disso, o aprendizado pode ser dificultado dependendo da distribuição dos dados. Para resolver este fenômeno, é utilizada a *Batch Normalization* (IOFFE e SZEGEDY, 2015), que escala a saída de cada camada da rede, visando padronização (dados com média  $\bar{X} = 0$  e desvio padrão  $\sigma = 1$ ) de cada variável de entrada, em cada *mini-batch*.

## 2.3 Aprendizado profundo para segmentação volumétrica

As redes neurais convolucionais têm sido aplicadas em tarefas de segmentação de imagens médicas 2D e obtido performances satisfatórias, com acurácia próxima da performance humana (RONNEBERGER *et al.*, 2015; HARIHARAN *et al.*, 2015; SEYEDHOSSEINI *et al.*,

2013). Isso desencadeou diversas tentativas de aplicação de CNNs sobre imagens médicas 3D, como as MRIs cerebrais contempladas neste trabalho. A adaptação das arquiteturas para imagens 3D se mostra uma tarefa desafiadora, porque dependendo da abordagem escolhida, o modelo pode ter alta perda de contexto e capacidade de generalização. Além disso, é importante considerar que dados volumétricos são usualmente maiores do que as imagens 2D, ocupando maior espaço na memória e tornando as regiões a serem segmentadas mais esparsas (especialmente no caso de imagens médicas).

Algumas abordagens misturam métodos de *shallow learning* com as CNNs 2D, mas neste trabalho serão utilizadas apenas abordagens totalmente *deep*. É possível escalar as camadas para convoluções 3D, *max pooling* 3D e outras operações que apenas recebam o *input* 3D e produzam uma segmentação adequada na imagem. A forma como as CNNs foram adequadas para imagens volumétricas será melhor detalhada nas descrições das arquiteturas escolhidas, na Seção 3.1.

# Capítulo 3

## Metodologia

A metodologia neste trabalho consiste em submeter os dados do BraTS e, a seguir, do FeTA a duas *pipelines*. A primeira *pipeline* é dada apenas pela utilização das redes neurais a serem definidas na Seção 3.1 para segmentação das regiões desejadas sobre as MRIs inteiras, sem pré-processamento. A segunda pipeline é melhor descrita na Seção 3.3. A segmentação então passa por um conjunto de validações, a fim de testar a eficiência da predição.

### 3.1 Arquiteturas de Redes Neurais

As cinco CNNs implementadas neste trabalho incluem as arquiteturas *Encoder-Decoder*: as U-Nets 3D (ÇIÇEK *et al.*, 2016) e as V-Nets (MILLETARI *et al.*, 2016); e as *Fully Convolutional Networks* (FCNs) (LONG *et al.*, 2015): HighRes3DNet (HR3N) (LI *et al.*, 2017), SkipDenseNet (SDN)(BUI *et al.*, 2017) e Med3D (S. CHEN *et al.*, 2019). Cada arquitetura será melhor descrita a seguir.

#### 3.1.1 U-Net 3D

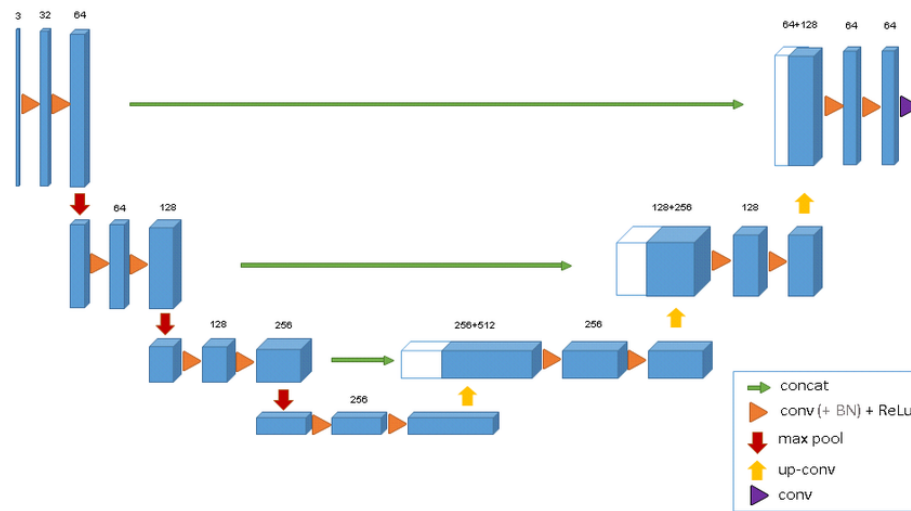
A U-Net 3D é uma adequação da arquitetura U-Net (RONNEBERGER *et al.*, 2015) para imagens 2D. A primeira U-Net foi desenvolvida utilizando a filosofia arquitetural *Encoder-Decoder*, que divide as camadas da rede neural em duas partes: a codificação, utilizada para analisar a imagem completa e então contraí-la nas dimensões espaciais — enquanto se adiciona mais canais — e a decodificação, que recebe a imagem contraída e a expande de modo a produzir uma segmentação com o tamanho original.

As operações realizadas em cada camada da U-Net 3D (ÇIÇEK *et al.*, 2016) seguem o mesmo modelo da U-Net original, adaptadas para imagens de três dimensões: convoluções 3D, *max pooling* 3D e convoluções transpostas 3D.

A arquitetura dessa rede neural consiste em quatro passos de codificação e mais quatro passos de decodificação. Para codificar, cada camada contém duas convoluções 3D com *kernel* de tamanho  $3 \times 3 \times 3$ , sendo que o resultado de cada convolução passa por uma *batch normalization* e pela função de ativação ReLU (KRIZHEVSKY *et al.*, 2012). Essa sequência de

convolução, *batch normalization* e ReLU é chamada de bloco convolucional. Entre cada um dos passos da codificação, também há um *max pooling* 3D com *kernel* de tamanho  $2 \times 2 \times 2$  e *stride* de 2 em cada dimensão. Para a decodificação, cada camada consiste de uma convolução transposta 3D (ou *upconvolution*) de tamanho  $2 \times 2 \times 2$  e *stride* de 2 em cada dimensão, precedido por dois blocos convolucionais com *kernel* de  $3 \times 3 \times 3$ . As conexões horizontais entre camadas de mesma resolução, chamadas comumente de *skip connections*, realizam uma concatenação das *features*. No final, é aplicada uma última convolução de  $1 \times 1 \times 1$  para transformar um espaço de *features* em um tensor contendo  $k$  *feature maps*, sendo  $k$  o número de classes do modelo. Assim, cada *feature map* prediz a probabilidade de uma classe a partir da aplicação da função *softmax* (melhor detalhada na Seção 3.2).

A arquitetura está esquematizada na Figura 3.1.



**Figura 3.1:** A arquitetura da U-Net 3D. As caixas azuis representam as *feature maps* e o número de canais está descrito sobre cada caixa. Figura extraída de (ÇIÇEK *et al.*, 2016).

### 3.1.2 V-Net

A arquitetura V-Net (MILLETARI *et al.*, 2016), como a U-Net 3D, se propõe a ser uma CNN totalmente tridimensional, com convoluções volumétricas em suas camadas. O formato da V-Net também é *Encoder-Decoder* com conexões residuais entre camadas de mesma resolução.

A V-Net original consiste de cinco passos de codificação e cinco de decodificação. No entanto, para este trabalho, foi necessário modificar de acordo com os recursos computacionais que tínhamos a disposição e à resolução espacial dos *patches* de entrada das redes. Então foram utilizados quatro *encoders* e quatro *decoders*. Cada camada da arquitetura contém de uma a três convoluções de tamanho  $3 \times 3 \times 3$  voxels. Além disso, nas camadas de *encoding* são realizadas operações de convolução para redução de resolução, que produzem efeitos semelhantes ao *pooling*, com tamanho  $2 \times 2 \times 2$  e *stride* de 2. Para as camadas de *decoding*, são realizadas operações de de-convolução com o mesmo tamanho, para aumentar a resolução.

Além das conexões residuais, cada camada conta com uma conexão residual que



encaminha a entrada das convoluções à saída da camada. Essas duas medidas colaboram com o aprendizado da função residual. Funções de ativação PReLU (ReLU paramétrica) são aplicadas por toda a arquitetura. Finalmente, a última camada da rede realiza uma convolução  $1 \times 1 \times 1$  para reduzir o número de canais de saída para a quantidade de classes a serem segmentadas e prever as probabilidades das classes com a função *softmax*. A Figura 3.2 ilustra a V-Net.

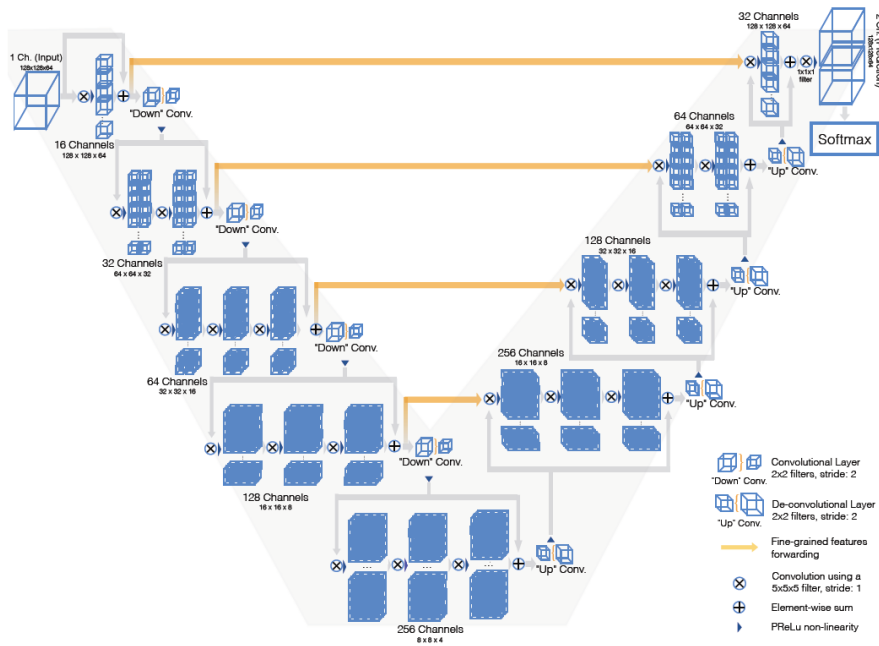


Figura 3.2: A arquitetura da V-Net. Figura extraída de (MILLETARI *et al.*, 2016).

### 3.1.3 HighResNet 3D

A arquitetura da HighResNet 3D (HR3N) (LI *et al.*, 2017) é proposta visando segmentação de alta resolução sobre imagens volumétricas, realizada de forma compacta. Diferentemente das *Encoder-Decoder*, a HR3N faz uso de convoluções dilatadas, evitando os efeitos de perda de resolução das operações de *up* e *down-sampling*. As convoluções dilatadas são operações convolucionais modificadas de modo a percorrer o *kernel* de forma diferente da feita em convoluções comuns. A modificação da operação é dada por um fator  $r$ , que define o quão dilatada ela será (é comum dizer que tal convolução é  $r$ -dilatada) (YU e KOLTUN, 2015).

Esta arquitetura conta com vinte camadas de blocos convolucionais de tamanho  $3 \times 3 \times 3$ , com blocos de conexões residuais a cada duas camadas convolucionais. Para a implementação neste projeto, foram utilizadas apenas três dessas camadas. A camada final de convolução  $1 \times 1 \times 1$  gera a predição junto a uma função *softmax*, assim como foi especificado nas arquiteturas anteriores. A Figura 3.3 detalha os blocos e camadas da arquitetura HR3N.

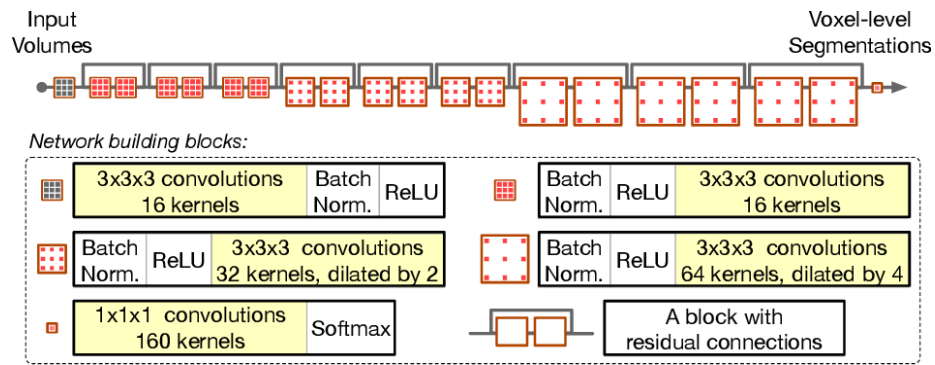


Figura 3.3: A arquitetura da HighRes3DNet. Figura extraída de (Li et al., 2017).

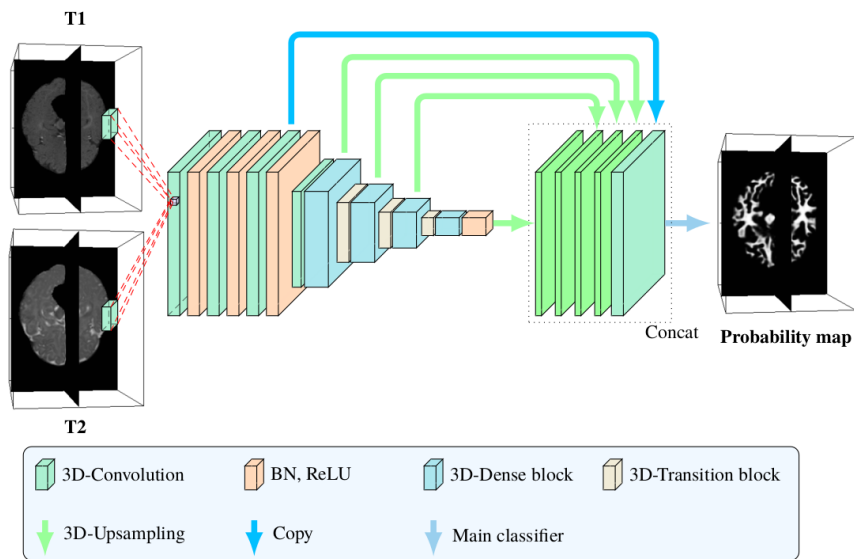
### 3.1.4 SkipDenseNet 3D

A SkipDenseNet 3D (SDN) (Bui et al., 2017) se utiliza da arquitetura da DenseNet (HUANG et al., 2017) para imagens volumétricas. As DenseNets (HUANG et al., 2017) são desenvolvidas principalmente para evitar o problema do *vanishing gradient*, que faz com que o gradiente tenda a zero conforme passa pelas camadas, e afete o aprendizado do modelo. Então, em vez de cada camada receber como entrada apenas a saída da camada anterior (como nas redes *feed-forward* tradicionais), há uma conexão residual para cada camada que concatena os *feature maps* de todas as camadas anteriores.

A SDN então constrói suas camadas se utilizando desse mecanismo de conexões densas. Na arquitetura original, são utilizadas 47 camadas, mas para este trabalho, foram utilizadas apenas 16. A arquitetura começa executando três camadas convolucionais extras para gerar entradas adequadas às camadas densas. Então, as 16 camadas se agrupam em quatro blocos densos. Cada bloco denso é definido por uma *batch normalization* e uma ReLU, seguidas de uma convolução  $1 \times 1 \times 1$ , novamente uma *batch normalization* e uma ReLU e finalmente uma convolução  $3 \times 3 \times 3$ . Entre os blocos densos existem blocos de transição para codificar (realizar *down-sampling*) os dados com convoluções de  $1 \times 1 \times 1$  e *stride* 2, e conexões densas para a decodificação (*up-sampling*). Para gerar a predição, a rede termina com uma convolução  $1 \times 1 \times 1$  dos dados de saída concatenados, semelhante ao final da execução das redes descritas nas subseções anteriores. A rede está esquematizada na Figura 3.4.

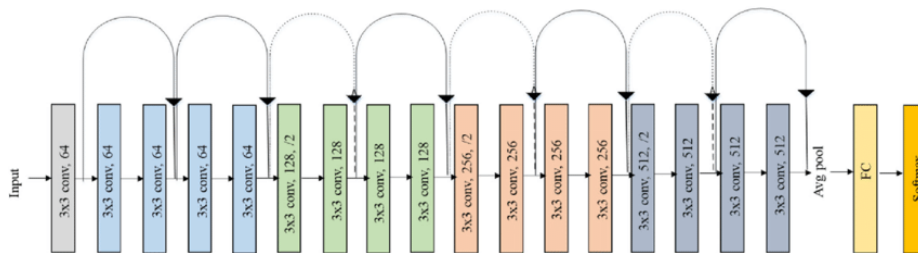
### 3.1.5 Med3D

A arquitetura da Med3D (S. CHEN et al., 2019) também utiliza o método *Encoder-Decoder* com conexões residuais. Mais especificamente, a codificação é feita utilizando a ResNet (HE et al., 2016), com pequenas modificações de modo que as convoluções sejam volumétricas e que as camadas se encaixem com a decodificação. No trabalho original, uma das *backbones* utilizadas foi a ResNet18 3D, ou seja, uma ResNet com 18 camadas convolucionais 3D, como exibido na Figura 3.5. Neste trabalho, foram utilizadas apenas oito. Assim, a codificação se dá por oito camadas de convolução  $3 \times 3 \times 3$ , com conexões residuais a cada quatro camadas. Cada camada também apresenta *batch normalization* e ReLU. Para a decodificação, a Med3D constrói oito ramificações, cada uma com camadas de *up-sampling* correspondentes à dimensão dos dados de entrada da mesma. As ramificações



**Figura 3.4:** Exemplo de aplicação da SkipDenseNet sobre MRIs cerebrais de contraste T1 e T2. Figura extraída de (Bui et al., 2017).

são úteis para diferenciar regiões de interesse nas imagens conforme o conjunto de dados utilizado.



**Figura 3.5:** Arquitetura da ResNet18, utilizada para codificação na Med3D. Note que a camada totalmente conectada (FC) no esquema é substituída pelas camadas de decodificação. Figura extraída de (RAMZAN et al., 2020).

Para este trabalho, cada ramificação de decodificação contém dois conjuntos de camadas compostas por uma convolução transposta de  $3 \times 3 \times 3$  e uma convolução comum de  $3 \times 3 \times 3$ . Finalmente, para obter a predição, como foi feito nas outras arquiteturas, cada ramificação aplica uma convolução de  $1 \times 1 \times 1$ .

## 3.2 Funções de Perda

Para a otimização dos pesos do *kernel* durante a execução das épocas das CNNs, é necessário definir uma função de perda. No caso de tarefas de segmentação altamente desbalanceadas, é interessante utilizar uma função de perda composta. Para os experimentos neste trabalho, a função de perda  $\mathcal{L}$  escolhida é dada pela composição da Entropia Cruzada (CE) com a *Dice Loss*:

$$\mathcal{L} = \mathcal{L}_{CE} + \mathcal{L}_{dice}. \quad (3.1)$$

As funções  $\mathcal{L}_{CE}$  e  $\mathcal{L}_{dice}$  serão melhor detalhadas nas próximas subseções.

Durante o cálculo das funções de perda, para obter a distribuição de probabilidade da predição, é utilizada a função de ativação *softmax*. Essa função normaliza a saída de uma camada para uma distribuição de probabilidade sobre as classes de saída preditas, e é definida por:

$$\text{softmax}(x_i) = \frac{\exp x_i}{\sum_j^K \exp x_j}, \quad \text{para } i = 1, \dots, K, \quad (3.2)$$

onde  $x_i$  é um elemento de  $\mathbf{x}$ , vetor de tamanho  $K$  a ser normalizado.

### 3.2.1 Entropia Cruzada

A medida de entropia cruzada quantifica a diferença entre duas distribuições de probabilidade. Para o cálculo de perda, a distribuição de probabilidade verdadeira  $p_i$  é dada pelos rótulos da imagem (alvo da predição) e a distribuição de probabilidade estimada  $q_i$  pela predição. A entropia cruzada é definida por:

$$\mathcal{L}_{CE} = - \sum_i^N p_i \log q_i, \quad (3.3)$$

onde  $N$  é a dimensão dos dados.

Em termos dos dados de entrada  $x$  e do alvo da predição  $y$ , a medida pode ser calculada da seguinte maneira:

$$\mathcal{L}_{CE}(x, y) = L = \{l_1, \dots, l_N\}, \quad l_n = -w_{y_n} \log \frac{\exp(x_{n,y_n})}{\sum_{c=1}^C \exp(x_{n,c})} \cdot 1, \quad (3.4)$$

onde  $w$  é o vetor de pesos e  $N$  a dimensão dos dados<sup>1</sup>. O vetor de pesos é pré-definido como hiperparâmetro do modelo, e é útil para conjuntos de treinamento desbalanceados — que é o caso das imagens médicas do projeto.

### 3.2.2 Dice Loss

A perda *Dice*, assim como a entropia cruzada, leva em conta as distribuições verdadeira ( $p_i$ ) e predita ( $q_i$ ). A ideia da função é maximizar o seguinte coeficiente:

$$\mathcal{L}_{dice} = \frac{2 \sum_i^N p_i q_i}{\sum_i^N p_i^2 + \sum_i^N q_i^2}, \quad (3.5)$$

onde  $N$  é a dimensão dos dados (MILLETARI *et al.*, 2016).

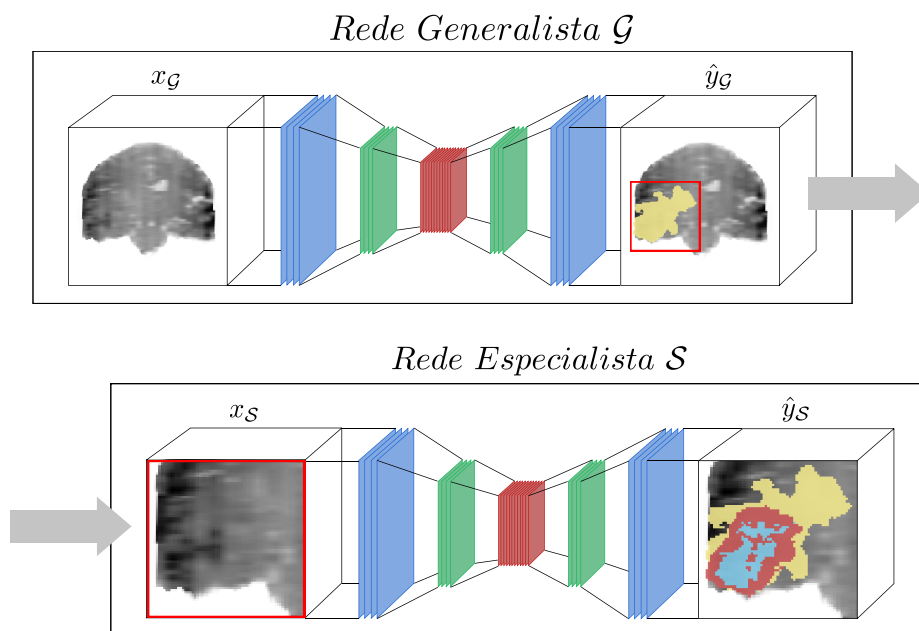
## 3.3 Redes Especialistas

A segunda *pipeline* é mais sofisticada do que a primeira: ela consiste em dois passos. Primeiro, a MRI é submetida à rede neural chamada generalista ( $\mathcal{G}$ ), na qual é feita uma

<sup>1</sup> <https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html>

predição grosseira binária de uma região de interesse. A partir dessa predição binária, é possível delimitar uma *bounding box* tomando os extremos da região predita. O segundo passo utiliza uma rede especialista ( $S$ ) para refinar as segmentações, submetendo as *bounding boxes* às redes neurais buscando uma segmentação multi-classe.

Para o conjunto de dados BraTS, a rede generalista deve segmentar um tumor inteiro. Assim, após esse resultado, a região de interesse (RoI) é gerada recortando a imagem ao redor do tumor detectado. Também é adicionado *padding* de 5 voxels em cada lado da RoI para evitar perda de contexto (para compensar as discrepâncias da predição de  $\mathcal{G}$ ), e a região de interesse é redimensionada para um tamanho padronizado (para que todo o conjunto de dados tenha mesmo tamanho). Finalmente, a rede especialista é executada sobre a RoI para segmentar as três sub-regiões do tumor (o *enhancing tumor* ET, o *tumor core* TC e o *whole tumor* WT) — que são melhor descritas na Seção 4.1. A Figura 3.6 esquematiza o funcionamento da rede especialista.



**Figura 3.6:** Pipeline proposta para a segmentação do conjunto de dados BraTS. A MRI inteira  $x_G$  é submetida à rede generalista  $\mathcal{G}$ , gerando a predição  $\hat{y}_G$ . Então, a RoI  $x_S$  obtida a partir da saída de  $\mathcal{G}$  é submetida à rede especialista  $\mathcal{S}$ , gerando a predição  $\hat{y}_S$ .

Já para o conjunto de dados FeTA, a rede generalista segmenta uma região do cérebro fetal que contém três estruturas escolhidas: o cerebelo, a substância cinzenta profunda e o tronco cerebral. A Seção 4.1 descreve em detalhes o conjunto de dados e as regiões anotadas clinicamente. A escolha dessa região se deu pela importância médica da mesma, pois são estruturas em rápido desenvolvimento em fetos e crianças, de modo que podem apresentar patologias (VIOLA *et al.*, 2011; GUI *et al.*, 2012). Além disso, as três regiões estão próximas uma da outra, então uma região de interesse que contém as três é relativamente pequena. Após o recorte da RoI, *padding* e redimensionamento (como feito nos dados do BraTS), a rede especialista segmenta as três estruturas cerebrais. O funcionamento da rede especialista está esquematizado na Figura 3.7.

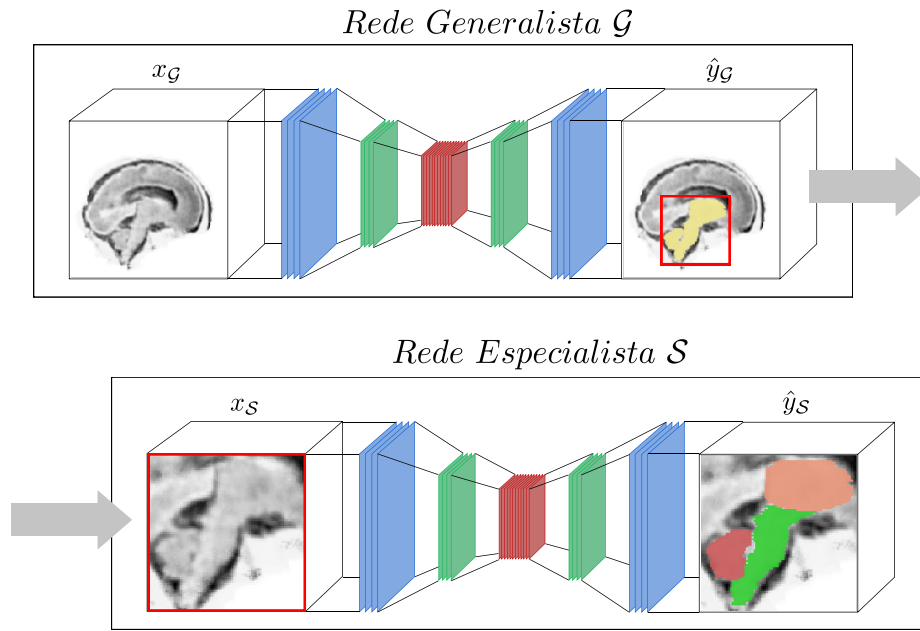


Figura 3.7: Pipeline proposta para a segmentação do conjunto de dados FeTA.

### 3.4 Testes

Após o treinamento das redes neurais sobre os conjuntos de dados selecionados, o modelo é testado sobre outra parte do conjunto de dados (um conjunto de teste). Para medir a eficácia do aprendizado, foram utilizadas as três métricas objetivas detalhadas na Seção 4.3. As previsões de segmentação também são salvas como rótulo das imagens, de modo que é possível visualizá-las após a execução. Algumas fatias das imagens geradas serão exibidas no Capítulo 5.

### 3.5 Implementação

Para o projeto todo, foi utilizada a linguagem de programação Python. A implementação das redes neurais utiliza a biblioteca Pytorch<sup>2</sup>, que contém classes e métodos adequadas para convoluções e outras operações detalhadas na Seção 2.1. Para o pré-processamento das imagens, foi utilizada a biblioteca Scikit-Image<sup>3</sup>. Finalmente, para o cálculo das métricas ao testar o modelo, foram utilizados os métodos e classes da biblioteca Scikit-Learn<sup>4</sup>.

O treinamento das redes foi feito utilizando duas GPUs de 8GB de memória. A quantidade de memória disponível limita o tamanho dos *mini-batches* utilizados, de modo que a escolha dos hiperparâmetros foi ajustada conforme essa informação.

<sup>2</sup> <https://pytorch.org>

<sup>3</sup> <https://scikit-image.org>

<sup>4</sup> <https://scikit-learn.org>

# Capítulo 4

## Setup Experimental

Para que a metodologia possa ser aplicada às tarefas de segmentação, é necessário definir um *setup* para os experimentos a serem executados. Neste projeto, dois conjuntos de dados diferentes serão utilizados para geração de predição. Para isso, foram determinados hiperparâmetros e configurações para o treinamento do modelo; além de protocolos e métricas de validação.

### 4.1 Conjuntos de dados

Com a abordagem de aprendizado profundo (com CNNs) sobre imagens volumétricas, é possível segmentar regiões de interesse em diferentes *datasets* de MRIs. Os conjuntos de dados dos desafios BraTS 2019 e FeTA 2021 utilizados neste trabalho serão melhor descritos a seguir.

#### 4.1.1 O conjunto de dados BraTS

O desafio BraTS utiliza o conjunto de dados clinicamente adquiridos que está disponível no portal do CBICA<sup>1</sup> (*Center for Biomedical Image Computing and Analytics*), da Universidade da Pensilvânia. São 335 MRIs de multi-contraste, de pacientes com glioma, dos quais 259 têm glioma de baixo grau (LGG) e 76 têm glioma de alto grau (HGG). Enquanto HGG são neoplasias cerebrais malignas com comportamento agressivo que não costumam ter respostas positivas ao tratamento oncológico, LGG são um pouco menos agressivos. O conjunto contém imagens de paciente pré e pós-tratamento.

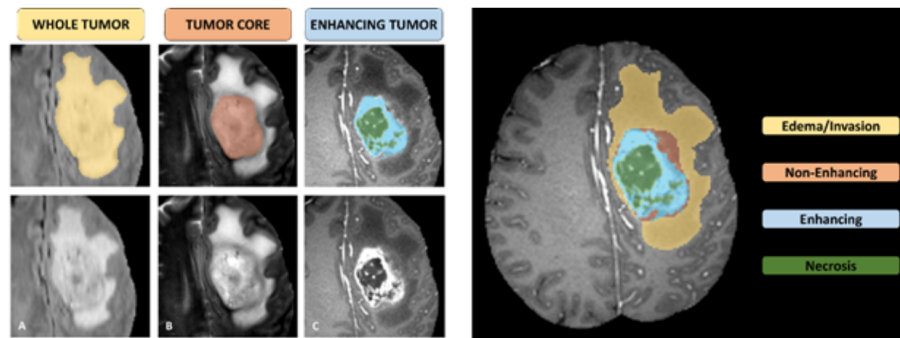
As MRIs foram obtidas em quatro centros diferentes: Universidade de Berna, Universidade de Debrecen, Universidade de Heidelberg e Hospital Geral de Massachusetts. Foram utilizadas *scanners* de diferentes fornecedores, forças de campo do ímã (1.5T e 3T) e implementações do sequenciamento das imagens (como 2D ou 3D). Todas as imagens têm volume  $240 \times 240 \times 155$  pixels, com resolução de  $1\text{ mm}^3$ . As ressonâncias magnéticas disponibilizadas são descritas por quatro diferentes tipos de contraste: “*native*” (T1),

---

<sup>1</sup> <https://ipp.cbica.upenn.edu>

“post-contrast T1-weighted” (T1Gd), “T2-weighted” (T2) e “T2 Fluid Attenuated Inversion Recovery” (FLAIR).

O desafio do BraTS consiste em utilizar a base de dados do CBICA para desenvolver um método que produza rótulos de segmentação de diferentes sub-regiões de gliomas. As imagens foram segmentadas manualmente por equipes de radiologistas. Assim, foram definidas sub-regiões consideradas para avaliação do desafio, são elas: o “*enhancing tumor*” (ET), o “*tumor core*” (TC) e o “*whole tumor*” (WT); que realçam ou atenuam certos padrões visuais nos diferentes contrastes do MRI. É possível ver a relação entre os diferentes contrastes e as regiões a serem segmentadas na Figura 4.1.



**Figura 4.1:** Sub-regiões do glioma. Em (A) é possível ver WT (amarelo) em T2-FLAIR, em (B) o TC (vermelho) visível em T2 e em (C) o ET (em azul) e o tecido necrótico, ambos visíveis em T1Gd. À direita, cada região segmentada é combinada na MRI. Figura extraída de (BAKAS, REYES *et al.*, 2018).

A escolha das classes se deu pela análise das anotações médicas. Apesar da padronização do processo de anotação médica, a classe que descreveria o tecido necrótico foi definida como tortuosa e pouco clara, por isso foi descartada para a segmentação automática. Assim, cada volume no formato de arquivo NIfTI (formato da *Neuroimaging Informatics Technology Initiative*) contém as 4 classes WT, ET, TC e o restante da imagem, o *background*.

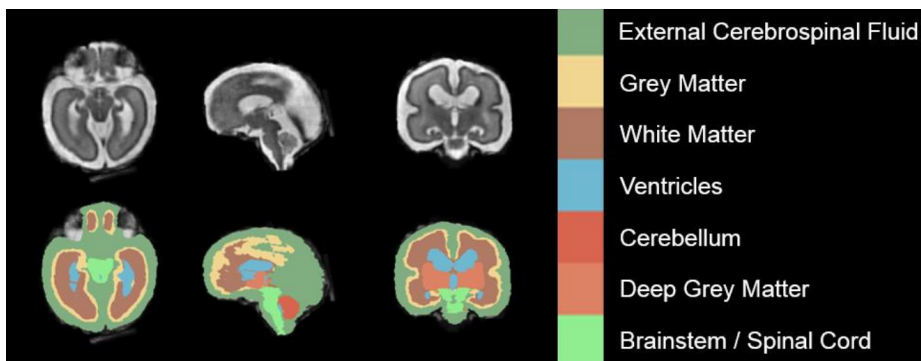
#### 4.1.2 O conjunto de dados FeTA

O conjunto de dados do desafio FeTA proposto pelo evento MICCAI em 2021 foi adquirido a partir de 50 mulheres grávidas no Hospital Infantil da Universidade de Zurique. São 80 MRIs no formato de arquivo NIfTI, descrevendo volumes *T2-weighted*. Todas as imagens têm volume  $256 \times 256 \times 256$  pixels, com resolução variável conforme tamanho do feto. Os *scanners* utilizados possuem força de campo do ímã de 1.5T ou 3T.

As imagens foram segmentadas manualmente por uma equipe de radiologistas. A anotação manual segmenta as principais estruturas cerebrais de fetos: líquido cefalorraquidiano externo (LCR), substância cinzenta (GM), substância branca (WM), ventrículos, cerebelo, substância cinzenta profunda (*deep GM*) e tronco cerebral (medula espinhal). Cada região pode ser melhor observada na Figura 4.2.

O desafio do FeTA consiste em utilizar esse *dataset* para desenvolver um método que produza rótulos de segmentação das diferentes regiões do cérebro dos fetos. As oito classes definidas para a segmentação automática são as sete regiões definidas acima, e o resto da imagem é rotulado como *background*.





**Figura 4.2:** Regiões cerebrais do feto a serem segmentadas. LCR externo em verde escuro, GM em amarelo, WM em marrom, ventrículos em azul, cerebelo em vermelho, deep GM em laranja e tronco cerebral em verde claro. Figura extraída de (PAYETTE *et al.*, 2021).

## 4.2 Hiperparâmetros

Para que a execução das *pipelines* seja feita corretamente, é necessário definir os parâmetros de treinamento e otimização. As configurações dos experimentos e os hiperparâmetros utilizados estão especificados nas tabelas 4.1, 4.2, 4.3 e 4.4.

Opções do Experimento	
exp_name	Nome do experimento
conv_name	Arquitetura
data_name	Nome do <i>dataset</i>
task_name	Tarefa de segmentação
num_classes	Número de classes a serem segmentadas

**Tabela 4.1:** Configurações que especificam as opções do experimento a ser realizado.

Parâmetros de Treinamento	
num_samples	Número de amostras utilizadas no treinamento
epoch_num	Número de épocas no treinamento
batch_size	Tamanho do <i>mini-batch</i>
test_freq	Testa a cada <i>test_freq</i> época

**Tabela 4.2:** Hiperparâmetros do treinamento.

Cada experimento se dá pela combinação de conjunto de dados (BraTS ou FeTA), arquitetura de rede neural e tarefa de segmentação. Nesta primeira *pipeline*, a única tarefa de segmentação é definida como “*all*”, que utiliza todos os rótulos presentes na anotação da imagem para a classificação das regiões segmentadas.

Para o carregamento do conjunto de dados, o algoritmo acessa o diretório *root* onde estão as imagens e seus respectivos rótulos, utilizando memória da GPU conforme o valor *num\_workers* definido.

Opções de Otimização	
lr	Taxa de aprendizado
weight_decay	Penalidade L2
momentum	<i>Momentum</i>
opt_step	Passo do otimizador
opt_gamma	$\gamma$ do otimizador

**Tabela 4.3:** Hiperparâmetros dos experimentos que dizem respeito ao otimizador.

Opções dos Dados	
root	Diretório raiz para os <i>datasets</i>
patching	Pré-processamento
num_workers	Número de <i>workers</i> no <i>dataloader</i>
w_size	Tamanho do eixo <i>x</i> para redimensionar imagem
h_size	Tamanho do eixo <i>y</i> para redimensionar imagem
z_size	Tamanho do eixo <i>z</i> para redimensionar imagem

**Tabela 4.4:** Configurações dos experimentos com respeito ao dataset.

Cada imagem passa por um pré-processamento dado por algoritmos de redimensionamento e amostragem. A definição do parâmetro `patching` como “*Resize*”, por exemplo, indica que a utilização de um algoritmo que redimensiona a imagem de maneira simples (podendo esticar os volumes, sem preservar proporções). O pré-processamento garante que todas as imagens processadas no experimento tenham tamanho `w_size × h_size × z_size` voxels.

Então a rede neural passa a buscar otimizar os pesos do *kernel* convolucional, utilizando `num_samples` para seu treinamento. Esse treinamento se dá executando o algoritmo definido pela rede neural `epoch_num` vezes, sendo que cada vez processa as imagens em “lotes” de tamanho `batch_size`. O otimizador em si pode ser configurado com alguns parâmetros como a taxa de aprendizado e a penalidade L2, que retardam o *learning* a fim de evitar *overfitting* do modelo. Além disso, o *momentum* regula a direção para qual o otimizador segue, conforme o gradiente descendente. A taxa de aprendizado de cada grupo de parâmetros decai de  $\gamma$  a cada `opt_step` épocas. A otimização dos pesos do *kernel* convolucional se dá a partir da função de perda  $\mathcal{L}$  definida na Seção 3.2, dada pela combinação das funções *Cross Entropy* e *Dice Loss*.

Todos os experimentos executados sobre os dados do BraTS e do FeTA definem os hiperparâmetros do otimizador com `lr = 0.001`, `weight_decay = 0.00005`, `momentum = 0.5`, `opt_step = 40` e `opt_gamma = 0.5`. Os parâmetros de treinamento são `epoch_num = 200`, `test_freq = 200` e `batch_size = 4`; com exceção da rede neural Med3D que, por causa das limitações da memória da GPU, utiliza `batch_size = 2`. Para as opções de pré-processamento e carregamento dos dados, define-se `num_workers = 8`, `patching` definido como *Resize* e dimensão de  $128 \times 128 \times 64$  voxels. As demais configurações são específicas de cada experimento, e serão melhor detalhadas no Capítulo 5.

## 4.3 Métricas

Como medidas objetivas para verificar a acurácias das predições, foram utilizadas três métricas: o coeficiente *Dice*, o Índice de Jaccard e a matriz de confusão. Elas serão melhor detalhadas nas subseções a seguir.

### 4.3.1 Coeficiente *Dice*

O coeficiente *dice* ou *F1 score* mede a proporção de acertos do modelo, e é calculado da seguinte maneira:

$$F1 = 2 \frac{PR}{P + R} = \frac{TP}{TP + \frac{1}{2}(FP + FN)}, \quad (4.1)$$

onde  $P$  é a precisão,  $R$  é o *recall* da predição,  $TP$  o número de verdadeiros positivos,  $FP$  o de falsos positivos e  $FN$  de falsos negativos <sup>2</sup>.

### 4.3.2 Índice de Jaccard

O índice de Jaccard obtém a proporção de intersecção sobre união (por isso também é conhecido como IoU, *intersection over union*), e pode ser descrito na forma:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}, \quad (4.2)$$

onde  $A$  é dado pelos rótulos da imagem (alvo da predição) e  $B$  é a segmentação predita. O índice também pode ser descrito em termos da precisão:

$$J = \frac{TP}{TP + FP + FN}, \quad (4.3)$$

onde  $TP$  é o número de verdadeiros positivos,  $FP$  o de falsos positivos e  $FN$  de falsos negativos.

### 4.3.3 Matriz de confusão

A matriz de confusão avalia a acurácia da classificação dispondo uma matriz  $C$  tal que  $C_{i,j}$  é o número de observações que estão na classe  $i$  e foram preditas como pertencentes à classe  $j$ .  $C$  tem dimensão  $k \times k$ , com  $k$  sendo o número de classes. Para este trabalho, a matriz de confusão é formata a fim de apresentar o valor relativo das predições em vez do valor absoluto (que seria em *voxels*).

<sup>2</sup> [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1\\_score.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html)



# Capítulo 5

## Resultados

Os experimentos descritos no Capítulo 4 foram realizados e os resultados são discutidos neste capítulo.

### 5.1 Execução sobre os dados do BraTS

A primeira *pipeline* é executada segmentando o conjunto de dados original com as cinco redes neurais escolhidas. Para o treinamento, utiliza-se `num_samples = 10`. O número de classes a ser segmentada é `num_classes = 4` e a `task_name` é `all`.

A Tabela 5.1 exibe as métricas *IoU* e *DSC* obtidas a partir da execução de cada uma das redes neurais para segmentar os dados.

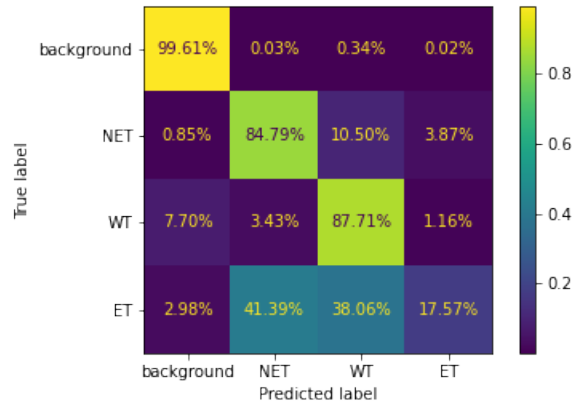
Métrica	V-Net	U-Net	HR3N	SDN	Med3D
DSC	0.5444	0.5828	0.49	0.6618	0.556
IoU	0.4583	0.4991	0.4091	0.5525	0.4491

**Tabela 5.1:** Métricas *DSC* e *IoU* da primeira *pipeline* executada sobre os dados do BraTS.

A matriz de confusão de uma das execuções destes experimentos está na Figura 5.1 e mostra que apenas o *background* e a região do WT foram segmentadas com maior acurácia, apesar de ocorrer confusões frequentes entre essas duas mesmas regiões.

Então, a segunda *pipeline* passa pelos dois passos definidos pelas redes generalista  $\mathcal{G}$  e especialista  $\mathcal{S}$ . Os parâmetros de treinamento de  $\mathcal{G}$  são os mesmos da primeira *pipeline*, exceto pelas configurações do experimento: `num_classes = 2` e `task_name` definida como `binary`.

A segmentação das imagens pela rede  $\mathcal{G}$  é utilizada para definição da RoI e recorte das MRIs. Esses volumes recortados então são submetidos à rede especialista, que já tem em seu pré-processamento o redimensionamento necessário. A seguir, a rede  $\mathcal{S}$  é treinada com `num_samples = 20`. O número de classes a ser segmentada é `num_classes = 4` e a `task_name` é `all`.



**Figura 5.1:** Matriz de confusão da primeira pipeline sobre os dados do BraTS, utilizando a arquitetura Med3D.

A Tabela 5.2 exibe as métricas *IoU* e *DSC* obtidas a partir da execução de cada uma das redes neurais para segmentar os dados, tanto para a rede generalista  $\mathcal{G}$  quanto para a rede especialista  $\mathcal{S}$ .

Métrica	Estratégia	V-Net	U-Net	HR3N	SDN	Med3D
DSC	$\mathcal{G}$	0.8132	0.5775	0.7498	0.8287	0.7159
	$\mathcal{S}$	0.6699	0.3103	0.3046	0.6404	0.3821
IoU	$\mathcal{G}$	0.6852	0.406	0.5998	0.7075	0.5575
	$\mathcal{S}$	0.5593	0.2681	0.266	0.5236	0.3061

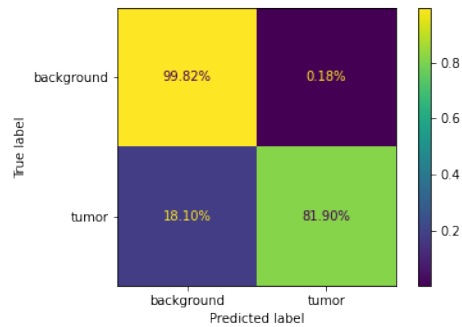
**Tabela 5.2:** Métricas *DSC* e *IoU* da segunda pipeline executada sobre os dados do BraTS, comparando as execuções de  $\mathcal{G}$  e  $\mathcal{S}$ .

As matrizes de confusão exibidas na Figura 5.2 mostram a alta acurácia na segmentação generalista do tumor em relação ao restante da imagem. Já no caso da rede especialista, o *background* e a região do WT foram segmentadas com maior acurácia.

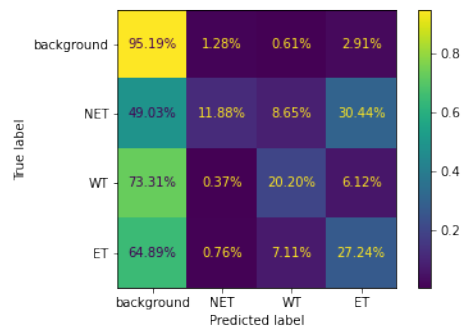
A Figura 5.3 mostra resultados qualitativos a partir de três amostras do conjunto de dados, suas respectivas RoIs em torno do WT, e as segmentações originais e previsões por modelo.

A segmentação dos tumores em relação às regiões saudáveis do cérebro foram as mais acuradas: tanto no primeiro *pipeline* quanto na delimitação da RoI nas redes generalistas, foram alcançadas previsões similares às anotações médicas. A detecção desses tumores então pode ser utilizada para análises médicas, para então se obter um diagnóstico levando em base medidas de volumetria.

No entanto, os resultados obtidos pelas redes especialistas foram menos acurados do que aqueles na primeira *pipeline*. Isso se dá pela alta variabilidade do tamanho dos tumores e da região do cérebro na qual eles se encontram. Isso demonstra que a abordagem especialista não foi adequada ao problema de segmentação de gliomas, e que o modelo da segunda *pipeline* precisa de ajustes.



(a) Matriz de confusão da execução generalista com a arquitetura SDN.



(b) Matriz de confusão da execução especialista com a arquitetura Med3D.

Figura 5.2: Matrizes de confusão da segunda pipeline sobre os dados do BraTS.

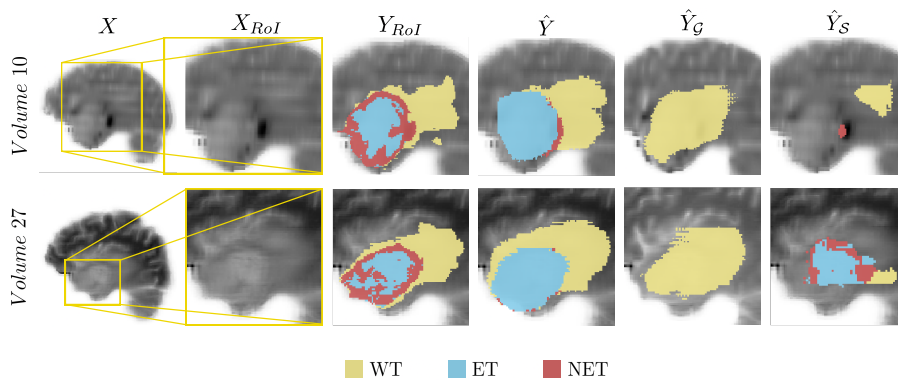


Figura 5.3: Segmentação das sub-regiões do glioma. As colunas indicam, em ordem, fatias sagitais da imagem  $X$ , a região de interesse  $X_{RoI}$ , o rótulo da região  $Y_{RoI}$  e as previsões  $\hat{Y}$  (da primeira pipeline),  $\hat{Y}_G$  (da rede generalista) e  $\hat{Y}_S$  (da rede especialista)

Comparativamente, a análise quantitativa dos dados mostra que as arquiteturas V-Net e SDN alcançaram maior acurácia em ambos os experimentos realizados. Isso se evidencia mais na execução das redes generalistas, que gerou *DSC* acima de 0.8 para essas arquiteturas.

Um problema evidenciado na análise qualitativa é a amostragem e as operações de redimensionamento das imagens ao executar o pré-processamento. Esse fenômeno poderia ser resolvido com outro processamento dos volumes de saída, a fim de refinar as predições. As etapas de pós-processamento, junto com possíveis ajustes no modelo especialista para se adequar ao conjunto BraTS, serão deixadas como trabalho futuro para o projeto.

## 5.2 Execução sobre os dados do FeTA

Novamente, a primeira *pipeline* é executada segmentando o conjunto de dados original com as cinco redes neurais escolhidas. Para o treinamento, utiliza-se `num_samples = 20`. O número de classes a ser segmentada é `num_classes = 8` e a `task_name` é `all`.

A Tabela 5.3 exibe as métricas *IoU* e *DSC* obtidas a partir da execução de cada uma das redes neurais para segmentar os dados.

Métrica	V-Net	U-Net	HR3N	SDN	Med3D
DSC	0.861	0.8355	0.8072	0.8749	0.7848
IoU	0.7635	0.7267	0.6884	0.7844	0.6586

**Tabela 5.3:** Métricas *DSC* e *IoU* da primeira *pipeline* executada sobre os dados do FeTA.

Na matriz de confusão exibida na Figura 5.4, é possível notar que as predições obtidas foram significativamente acuradas. Especificamente o *background* e as estruturas do LCR externo, da substância cinzenta (GM) e da substância branca (WM) foram segmentadas com maior acurácia dentre as regiões de interesse.

A segunda *pipeline* passa pelas redes generalista  $\mathcal{G}$  e a especialista  $\mathcal{S}$ . Os parâmetros de treinamento de  $\mathcal{G}$  são os mesmos da primeira *pipeline*, exceto pelas configurações do experimento: o número de classes a ser segmentada é `num_classes = 2` e a `task_name` é `binary`.

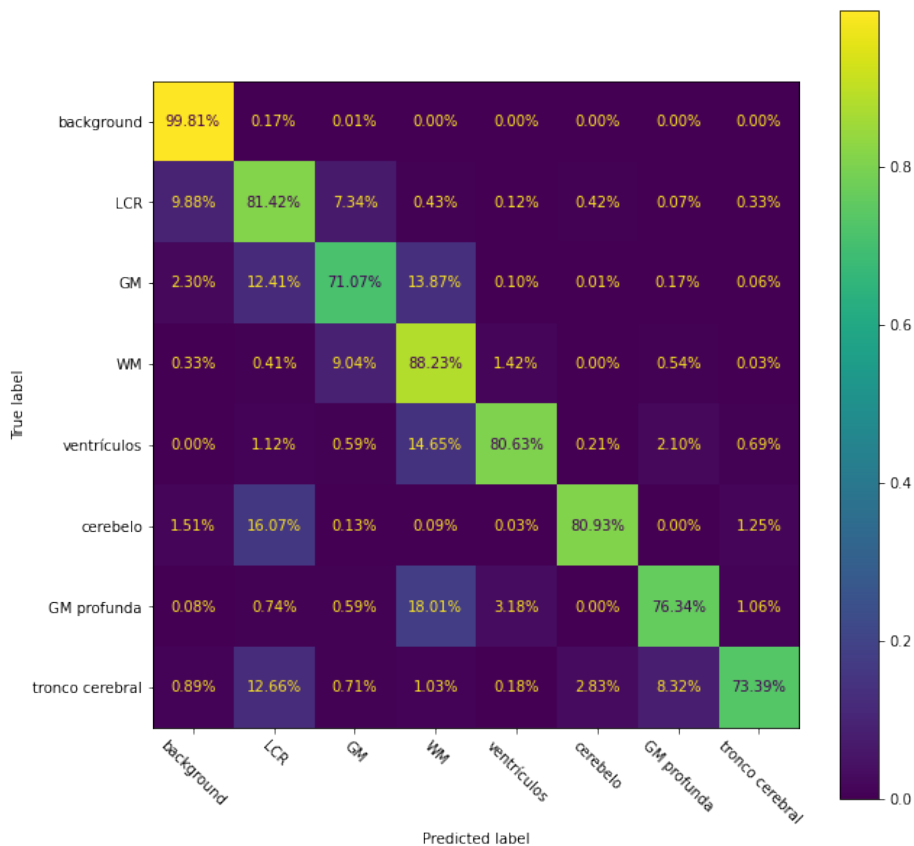
Após o recorte das MRIs conforme as RoI (definidas pela segmentação em  $\mathcal{G}$ ), os volumes recortados são submetidos à rede especialista, que já tem em seu pré-processamento o redimensionamento necessário. A rede  $\mathcal{S}$  é então treinada com `num_samples = 20`. O número de classes a ser segmentada é `num_classes = 4` e a `task_name` é `specialist`.

A Tabela 5.4 exibe as métricas *IoU* e *DSC* obtidas nos testes da segmentação dos dados utilizando cada uma das redes neurais, tanto para a rede generalista  $\mathcal{G}$  quanto para a rede especialista  $\mathcal{S}$ .

As matrizes de confusão exibidas na Figura 5.5 mostram a alta acurácia na segmentação generalista das três regiões em relação ao restante da imagem. Por sua vez, a rede especialista revela maior acurácia na detecção entre o *background* e a região de GM profunda, apesar de ocorrer confusões frequentes entre essas duas mesmas regiões. Também é



## 5.2 | EXECUÇÃO SOBRE OS DADOS DO FETA

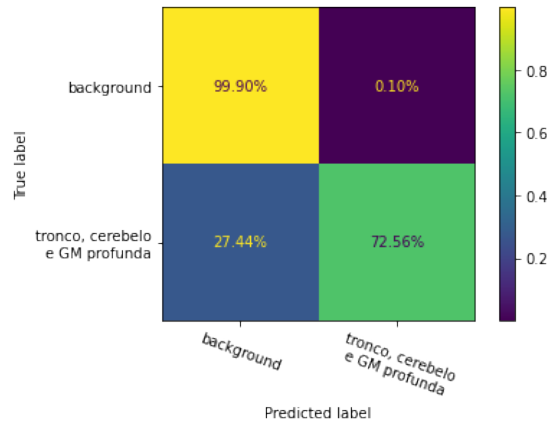


**Figura 5.4:** Matriz de confusão da primeira pipeline sobre os dados do FeTA, utilizando a arquitetura U-Net.

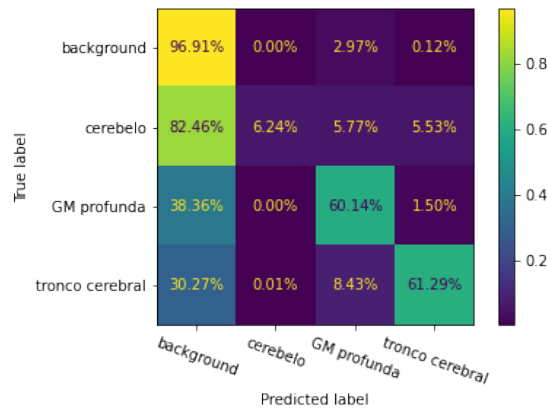
Métrica	Estratégia	V-Net	U-Net	HR3N	SDN	Med3D
DSC	$\mathcal{G}$	0.8864	0.8094	0.7465	0.7711	0.8119
	$\mathcal{S}$	0.8996	0.8363	0.5905	0.9011	0.9313
IoU	$\mathcal{G}$	0.7959	0.6798	0.5955	0.6275	0.6833
	$\mathcal{S}$	0.8233	0.7296	0.4866	0.8263	0.8741

**Tabela 5.4:** Métricas DSC e IoU da segunda pipeline executada sobre os dados do FeTA, comparando as execuções de  $\mathcal{G}$  e  $\mathcal{S}$ .

possível notar que voxels de *background* são frequentemente classificados (erroneamente) como tronco cerebral.



(a) Matriz de confusão da execução generalista com a arquitetura U-Net.



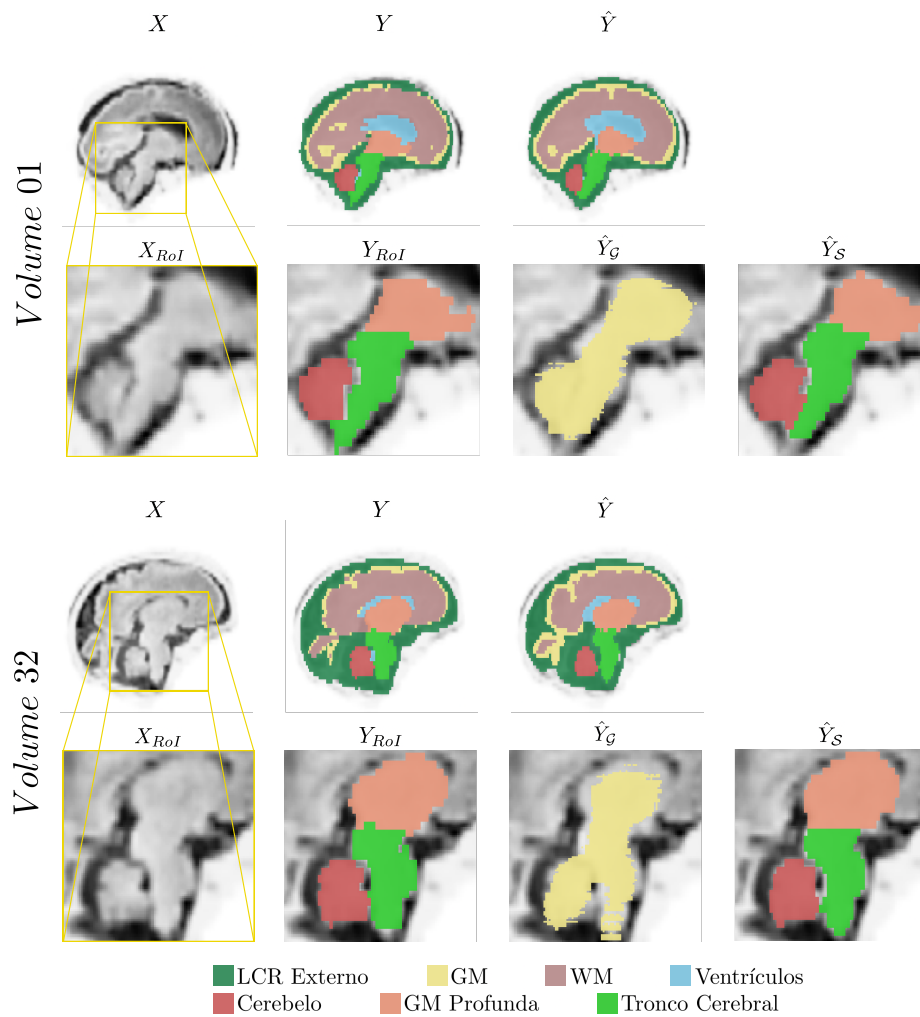
(b) Matriz de confusão da execução especialista com a arquitetura HR3N.

**Figura 5.5:** Matrizes de confusão da segunda pipeline sobre os dados do FeTA.

A Figura 5.6 mostra resultados qualitativos a partir de três amostras do conjunto de dados, suas respectivas RoIs em torno do cerebello, substância cinzenta profunda (apesar de ser frequentemente confundida com o *background*) e tronco cerebral. Além disso, o esquema exhibe as segmentações originais e predições por modelo.

Os resultados obtidos a partir dos experimentos sobre os dados do FeTA foram mais acurados do que aqueles nos experimentos sobre o BraTS, por causa da maior padronização entre os volumes (as estruturas cerebrais não apresentam alta variabilidade de tamanho e posição). Já na primeira *pipeline* foi possível notar alta acurácia da predição em todas as arquiteturas, com *DSC* em torno de 0.8.

O fato das três regiões a serem segmentadas (tronco cerebral, cerebello e substância cinzenta profunda) estarem relativamente distantes umas das outras piorou o desempenho da rede generalista. Por isso, a maioria das RoIs delimitadas por essa etapa não contém todas as estruturas que precisavam ser segmentadas. No entanto, ao segmentar apenas as estruturas dentro da RoI, as redes especialistas alcançaram alto desempenho em relação



**Figura 5.6:** Segmentação das sub-regiões do cérebro fetal. As colunas indicam, em ordem, fatias sagitais da imagem  $X$ , o rótulo  $Y$  e as previsões  $\hat{Y}$ ; uma para cada pipeline.

ao que tinha sido obtido nos dados do BraTS. Exceto pela HR3N, todas as arquiteturas tiveram maior  $DSC$  do que o alcançado na primeira *pipeline*, e as redes SDN e Med3D atingiram  $DSC$  superior a 0.9.

O problema das operações de redimensionamento se repete nestes experimentos, de modo que também precisa de pós-processamento para refinar as previsões. Também é interessante executar experimentos de redes especialistas sobre as outras regiões do cérebro (líquido cefalorraquidiano externo, substância cinzenta, substância branca e ventrículos). Essas etapas são sugeridas como tarefas futuras para este projeto.



# Capítulo 6

## Conclusão

A aplicação de redes neurais convolucionais ao problema de segmentação de imagens médicas tem obtido resultados altamente acurados, similares às anotações médicas. Neste trabalho, o uso das arquiteturas U-Net 3D, V-Net, HighRes3DNet, SkipDenseNet e Med3D sobre dois conjuntos de dados distintos alcançou predições com acurácias variáveis conforme método de segmentação e complexidade do conjunto.

A segmentação de tumores no conjunto de dados BraTS obteve acurácia satisfatória (em relação às anotações médicas), principalmente sem a utilização das redes especialistas, que se mostraram pouco adequadas ao problema da detecção de gliomas.

A seguir, foram realizados experimentos de segmentação automática sobre o conjunto de dados FeTA, com o objetivo de segmentar estruturas cerebrais em MRIs fetais. Neste caso, os resultados obtidos se mostraram mais acurados que os anteriores, principalmente com a utilização de uma *pipeline* com redes generalista e especialista. A delimitação de uma região de interesse ao redor de algumas estruturas se mostra particularmente relevante quando a tarefa consiste em segmentar essas estruturas.

Assim, é possível concluir que a extensão de modelos de segmentação de estruturas cerebrais adultas para MRIs fetais não é realizada trivialmente. É necessária uma abordagem específica sobre essas imagens, e a literatura sobre esse assunto é escassa. Para isso, a utilização das redes especialistas se mostrou útil e obteve resultados acurados. Portanto, a aplicação das redes especialistas sobre outras estruturas cerebrais seria interessante e poderá ser realizada no futuro dentro deste projeto. Análises de volumetria sobre as segmentações obtidas se mostram necessárias para interpretação e uso médico das predições. Essa etapa e demais ajustes aos modelos a fim de aperfeiçoá-los também deverão ser implementados no futuro.

Outros trabalhos também poderão ser desenvolvidos tomando como base os resultados obtidos neste projeto. Para o futuro, um desafio interessante é utilizar a segmentação de MRIs fetais para detecção de patologias, com a extração de métricas objetivas para a comparação entre volumes saudáveis e patológicos. Outro trabalho relevante para o futuro é o estudo de MRIs pediátricas, com foco na segmentação e detecção de patologias nas imagens.



## Referências

- [BAKAS, AKBARI *et al.* 2017] Spyridon BAKAS, Hamed AKBARI *et al.* “Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features”. Em: *Scientific data* 4.1 (2017), pgs. 1–13 (citado na pg. 1).
- [BAKAS, REYES *et al.* 2018] Spyridon BAKAS, Mauricio REYES *et al.* “Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge”. Em: *arXiv preprint arXiv:1811.02629* (2018) (citado nas pgs. 1, 18).
- [BOUGACHA *et al.* 2018] Aymen BOUGACHA *et al.* “Comparative study of supervised and unsupervised classification methods: application to automatic mri glioma brain tumors segmentation”. Em: *2018 4th International Conference on advanced technologies for signal and image processing (ATSIP)*. IEEE. 2018, pgs. 1–5 (citado na pg. 5).
- [BUI *et al.* 2017] Toan Duc BUI, Jitae SHIN e Taesup MOON. “3d densely convolutional networks for volumetric segmentation”. Em: *arXiv preprint arXiv:1709.03199* (2017) (citado nas pgs. 9, 12, 13).
- [CARDENAS *et al.* 2020] Agustin M CARDENAS, Matthew T WHITEHEAD e Dorothy I BULAS. “Fetal neuroimaging update”. Em: *Seminars in pediatric neurology*. Vol. 33. Elsevier. 2020, pg. 100801 (citado na pg. 2).
- [S. CHEN *et al.* 2019] Sihong CHEN, Kai MA e Yefeng ZHENG. “Med3d: transfer learning for 3d medical image analysis”. Em: *arXiv preprint arXiv:1904.00625* (2019) (citado nas pgs. 9, 12).
- [W. CHEN *et al.* 2017] Wei CHEN *et al.* “Automatic brain tumor segmentation based on features of separated local square”. Em: *2017 Chinese automation Congress (CAC)*. IEEE. 2017, pgs. 6489–6493 (citado na pg. 5).
- [ÇIÇEK *et al.* 2016] Özgün ÇIÇEK, Ahmed ABDULKADIR, Soeren S LIENKAMP, Thomas BROX e Olaf RONNEBERGER. “3d u-net: learning dense volumetric segmentation from sparse annotation”. Em: *International conference on medical image computing and computer-assisted intervention*. Springer. 2016, pgs. 424–432 (citado nas pgs. 9, 10).

- [DESPOTOVIĆ *et al.* 2015] Ivana DESPOTOVIĆ, Bart GOOSSENS e Wilfried PHILIPS. “Mri segmentation of the human brain: challenges, methods, and applications”. Em: *Computational and mathematical methods in medicine 2015* (2015) (citado na pg. 1).
- [DHANACHANDRA *et al.* 2015] Nameirakpam DHANACHANDRA, Khumanthem MANGLEM e Yambem Jina CHANU. “Image segmentation using k-means clustering algorithm and subtractive clustering algorithm”. Em: *Procedia Computer Science* 54 (2015), pgs. 764–771 (citado na pg. 5).
- [DUMOULIN e VISIN 2016] Vincent DUMOULIN e Francesco VISIN. “A guide to convolution arithmetic for deep learning”. Em: *arXiv preprint arXiv:1603.07285* (2016) (citado na pg. 6).
- [GLOROT *et al.* 2011] Xavier GLOROT, Antoine BORDES e Yoshua BENGIO. “Deep sparse rectifier neural networks”. Em: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop e Conference Proceedings. 2011, pgs. 315–323 (citado na pg. 7).
- [GUI *et al.* 2012] Laura GUI *et al.* “Morphology-driven automatic segmentation of mr images of the neonatal brain”. Em: *Medical image analysis* 16.8 (2012), pgs. 1565–1579 (citado na pg. 15).
- [HARIHARAN *et al.* 2015] Bharath HARIHARAN, Pablo ARBELÁEZ, Ross GIRSHICK e Jitendra MALIK. “Hypercolumns for object segmentation and fine-grained localization”. Em: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pgs. 447–456 (citado na pg. 7).
- [HATAMI *et al.* 2019] Toktam HATAMI, Mohammad HAMGHALAM, Omid REYHANI-GALANGASHI e Sattar MIRZAKUCHAKI. “A machine learning approach to brain tumors segmentation using adaptive random forest algorithm”. Em: *2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI)*. IEEE. 2019, pgs. 076–082 (citado na pg. 5).
- [HE *et al.* 2016] Kaiming HE, Xiangyu ZHANG, Shaoqing REN e Jian SUN. “Deep residual learning for image recognition”. Em: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pgs. 770–778 (citado na pg. 12).
- [HUANG *et al.* 2017] Gao HUANG, Zhuang LIU, Laurens VAN DER MAATEN e Kilian Q WEINBERGER. “Densely connected convolutional networks”. Em: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pgs. 4700–4708 (citado na pg. 12).
- [IOFFE e SZEGEDY 2015] Sergey IOFFE e Christian SZEGEDY. “Batch normalization: accelerating deep network training by reducing internal covariate shift”. Em: *International conference on machine learning*. PMLR. 2015, pgs. 448–456 (citado na pg. 7).



- [KRIZHEVSKY *et al.* 2012] Alex KRIZHEVSKY, Ilya SUTSKEVER e Geoffrey E HINTON. “Imagenet classification with deep convolutional neural networks”. Em: *Advances in neural information processing systems* 25 (2012), pgs. 1097–1105 (citado nas pgs. 2, 6, 9).
- [LECUN *et al.* 1998] Yann LECUN, Léon BOTTOU, Yoshua BENGIO e Patrick HAFFNER. “Gradient-based learning applied to document recognition”. Em: *Proceedings of the IEEE* 86.11 (1998), pgs. 2278–2324 (citado na pg. 6).
- [LI *et al.* 2017] Wenqi LI *et al.* “On the compactness, efficiency, and representation of 3d convolutional networks: brain parcellation as a pretext task”. Em: *International conference on information processing in medical imaging*. Springer. 2017, pgs. 348–360 (citado nas pgs. 9, 11, 12).
- [LONG *et al.* 2015] Jonathan LONG, Evan SHELHAMER e Trevor DARRELL. “Fully convolutional networks for semantic segmentation”. Em: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pgs. 3431–3440 (citado na pg. 9).
- [MENZE *et al.* 2014] Bjoern H MENZE *et al.* “The multimodal brain tumor image segmentation benchmark (brats)”. Em: *IEEE transactions on medical imaging* 34.10 (2014), pgs. 1993–2024 (citado na pg. 1).
- [MILLETARI *et al.* 2016] Fausto MILLETARI, Nassir NAVAB e Seyed-Ahmad AHMADI. “V-net: fully convolutional neural networks for volumetric medical image segmentation”. Em: *2016 fourth international conference on 3D vision (3DV)*. IEEE. 2016, pgs. 565–571 (citado nas pgs. 9–11, 14).
- [PAYETTE *et al.* 2021] Kelly PAYETTE *et al.* “An automatic multi-tissue human fetal brain segmentation benchmark using the fetal tissue annotation dataset”. Em: *Scientific Data* 8.1 (2021), pgs. 1–14 (citado na pg. 19).
- [POHLE e TOENNIES 2001] Regina POHLE e Klaus D TOENNIES. “Segmentation of medical images using adaptive region growing”. Em: *Medical Imaging 2001: Image Processing*. Vol. 4322. International Society for Optics e Photonics. 2001, pgs. 1337–1346 (citado na pg. 5).
- [RAMZAN *et al.* 2020] Farheen RAMZAN *et al.* “A deep learning approach for automated diagnosis and multi-class classification of alzheimer’s disease stages using resting-state fmri and residual neural networks”. Em: *Journal of medical systems* 44.2 (2020), pgs. 1–16 (citado na pg. 13).
- [RONNEBERGER *et al.* 2015] Olaf RONNEBERGER, Philipp FISCHER e Thomas BROX. “U-net: convolutional networks for biomedical image segmentation”. Em: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pgs. 234–241 (citado nas pgs. 7, 9).

- [SEYEDHOSSEINI *et al.* 2013] Mojtaba SEYEDHOSSEINI, Mehdi SAJJADI e Tolga TASHIZEN. “Image segmentation with cascaded hierarchical models and logistic disjunctive normal networks”. Em: *Proceedings of the IEEE international conference on computer vision*. 2013, pgs. 2168–2175 (citado na pg. 7).
- [VIOLA *et al.* 2011] A VIOLA *et al.* “Is brain maturation comparable in fetuses and premature neonates at term equivalent age?” Em: *American journal of neuroradiology* 32.8 (2011), pgs. 1451–1458 (citado na pg. 15).
- [YU e KOLTUN 2015] Fisher YU e Vladlen KOLTUN. “Multi-scale context aggregation by dilated convolutions”. Em: *arXiv preprint arXiv:1511.07122* (2015) (citado na pg. 11).